

EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEE	RRR	FFF
EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEEEEEEEE	RRRRRRRRR	FFFFFFFFFF
EEE	RRR RRR	FFF
EEEEEEEEE	RRR	FFF
EEEEEEEEE	RRR	FFF
EEEEEEEEE	RRR	FFF

FILEID**SBI

C 1

ER
PS

PS
--
SE

Ph
--
in
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
79
Th
15
1

Ma
--
S
0
Th
MA

The image shows a 16x16 grid of binary symbols (0s and 1s) representing a convolutional neural network's feature map. The symbols are arranged in a pattern where they form a large cross shape in the center, with vertical and horizontal bars extending from it. The symbols are grouped into four main quadrants: top-left (SS), top-right (BB), bottom-left (LL), and bottom-right (II). The symbols are represented by black dots on a white background.

```
0001 C
0002 C Version: 'V04-000'
0003 C
0004 C*****
0005 C*
0006 C* (COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0007 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0008 C* ALL RIGHTS RESERVED.
0009 C*
0010 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0011 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0012 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0013 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0014 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0015 C* TRANSFERRED.
0016 C*
0017 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0018 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0019 C* CORPORATION.
0020 C*
0021 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0022 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0023 C*
0024 C*
0025 C*****
0026 C
0027
0028 SUBROUTINE SBI (LUN)
0029
0030 C
0031 C AUTHOR BRIAN PORTER CREATION DATE 27-AUG-1979
0032 C
0033 C++
0034 C Functional description:
0035 C
0036 C Modified by:
0037 C
0038 C V03-003 EAD0001 Elliott A. Drayton 18-Feb-1984
0039 C Add UVAX-1 support.
0040 C
0041 C V03-002 SAR0096 Sharon A. Reynolds, 20-jun-1983
0042 C Changed the carriage control in the 'format' statements
0043 C for use with ERF.
0044 C
0045 C v03-001 BP0001 Brian Porter, 05-APR-1982
0046 C Corrected sbi alert bug.
0047 C++
0048 C--
0049
0050
0051 C INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0110 C INCLUDE 'SRC$:SYECOM.FOR /NOLIST'
0238
0239
0240
0241
0242
```

```
0243      BYTE          LUN
0244
0245      INTEGER*4    SBI_FAULT
0246
0247      INTEGER*4    SBI_COMP
0248
0249      INTEGER*4    SBI_MAINT
0250
0251      INTEGER*4    SBI_ERR
0252
0253      INTEGER*4    SBI_TO
0254
0255      INTEGER*4    SILO(0:15)
0256
0257      INTEGER*4    SBI_REGA(0:15)
0258
0259      INTEGER*4    ERROR_PC_780
0260
0261      INTEGER*4    ERROR_PSL_780
0262
0263      integer*4    error_pc_750
0264
0265      integer*4    error_psl_750
0266
0267      INTEGER*4    FIELD
0268
0269      EQUIVALENCE   (SBI_FAULT,EMB(16))
0270
0271      EQUIVALENCE   (SBI_COMP,EMB(20))
0272
0273      EQUIVALENCE   (SBI_MAINT,EMB(24))
0274
0275      EQUIVALENCE   (SBI_ERR,EMB(28))
0276
0277      EQUIVALENCE   (SBI_TO,EMB(32))
0278
0279      EQUIVALENCE   (SILO,EMB(36))
0280
0281      EQUIVALENCE   (SBI_REGA,EMB(100))
0282
0283      EQUIVALENCE   (ERROR_PC_780,EMB(164))
0284
0285      EQUIVALENCE   (ERROR_PSL_780,EMB(168))
0286
0287      equivalence    (error_pc_750,emb(16))
0288
0289      equivalence    (error_psl_750,emb(20))
0290
0291      integer*4     memory_registers_uv1(0:4)
0292
0293      equivalence    (memory_registers_uv1(0),emb(16))
0294
0295      PARAMETER     ASYNC_WRITE = 7
0296
0297      integer*4     compress4
0298
0299      logical*1     diagnostic_mode
```

```
0300  
0301  
0302 CALL FRCTOF (LUN)  
0303  
0304 call header (lun)  
0305  
0306  
0307 c  
0308 c 11/780, 782, 785 support  
0309  
0310  
0311 if (  
0312 1 lib$extzv(24,8,emb$l_hd_sid) .eq. 255  
0313 1 or.  
0314 1 lib$extzv(24,8,emb$l_hd_sid) .eq. 1  
0315 1 ) then  
0316 if (emb$w_hd_entry .eq. '07'x) then  
0317 call logger (lun,'ASYNCHRONOUS WRITE')  
0318 else  
0319 call logger (lun,'SBI FAULT')  
0320 endif  
0321  
0322 call linchk (lun,2)  
0323  
0324  
0325 10 write(lun,10) error_pc_780  
0326 format('/',t8,'ERROR PC',t24,z8.8)  
0327 call vaxpsl (lun,error_psl_780)  
0328 diagnostic_mode = .false.  
0329  
0330  
0331 if (iand(sbi_maint,'f05ff900'x) .ne. 0) diagnostic_mode = .true.  
0332  
0333 if (.not. diagnostic_mode) then  
0334 CALL SBI_FAULTREG (LUN,SBI_FAULT)  
0335 CALL SBI_COMPARATOR (LUN,SBI_COMP)  
0336 CALL SBI_MAINTENANCE (LUN,SBI_MAINT)  
0337 CALL SBI_ERROR (LUN,SBI_ERR)  
0338 CALL SBI_TIMEOUT (LUN,SBI_TO)  
0339 else  
0340  
0341 call linchk (lun,6)  
0342  
0343  
0344 28 write(lun,28) sbi_fault,sbi_comp,sbi_maint,sbi_err,sbi_to  
0345 format(' ',t8,'SBIFS',t24,z8.8,/,  
0346 1 t8,'SBISCI',t24,z8.8,/,  
0347 1 t8,'SBIMT',t24,z8.8,/,  
0348 1 t40,'DIAGNOSTIC MODE',/,  
0349 1 t8,'SBIER',t24,z8.8,/,  
0350 1 t8,'SBITA',t24,z8.8,/,  
0351  
0352  
0353  
0354  
0355  
0356
```

```
0357      endif
0358
0359      IF (JIAND(SBI_COMP,'A0000000'X) .NE. 0
0360      1 OR.
0361      2 JIAND(SBI_FAULT,'10000'X) .NE. 0) THEN
0362
0363      if (.not. diagnostic_mode) then
0364
0365          CALL LINCHK (LUN,3)
0366
0367      30      WRITE(LUN,30)
0368      FORMAT(/' ','SBI SILO LOCKED, DETAILED SUMMARY',/)
0369
0370      DO 50,I = 0,15
0371
0372          CALL SBI_SILO (LUN,SILO(I))
0373
0374      50      CONTINUE
0375      else
0376
0377          CALL LINCHK (LUN,3)
0378
0379      52      WRITE(LUN,52)
0380      FORMAT(/' ','SBI SILO LOCKED',/)
0381
0382      do 54,i = 0,15
0383
0384          call lincbk (lun,1)
0385
0386      53      write(lun,53) silo(i)
0387      format(' ',t24,z8.8)
0388
0389      54      continue
0390      endif
0391      ENDIF
0392
0393      DO 80,I = 0,15
0394
0395      IF (SBI_REGA(I) .NE. 0) THEN
0396
0397          CALL LINCHK (LUN,2)
0398
0399      55      WRITE(LUN,55) I
0400      FORMAT(/' ','ADAPTER TR# ',I<compress4 (i)>,'.')
0401
0402      CALL CLASSIFY (LUN,SBI_REGA(I))
0403      ENDIF
0404
0405      80      CONTINUE
0406
0407      c
0408      c      11/750 support
0409      c
0410
0411      else if (lib$extzv(24,8,emb$1_hd_sid) .eq. 2) then
0412          if (emb$w_hd_entry .eq. '07'x) then
```

```
0414
0415     call logger (lun,'WRITE BUS ERROR')
0416   endif
0417
0418   write(lun,10) error_pc_750
0419
0420   call vaxpsl (lun,error_psl_750)
0421
0422   c
0423   c    UVAX-1 support
0424   c
0425
0426   else if (lib$extzv(24,8,emb$1_hd_sid) .eq. 7) then
0427
0428     if (emb$w_hd_entry .eq. 7) then
0429
0430       call logger (lun,'ASYNCHRONOUS WRITE')
0431
0432       do 85,i = 1,16
0433
0434       if (lib$extzv(15,1,memory_registers_uv1(i)) .eq. 1) then
0435
0436         call memory_register_uv1 (lun,memory_registers_uv1)
0437       endif
0438
0439   85   continue
0440
0441   endif
0442
0443   c
0444   c    The IF-THEN-ELSE must be expanded at this point
0445   c    to provide additional CPU "ASYNCHRONOUS WRITE"
0446   c    ERROR" support.
0447   c
0448
0449   endif
0450
0451   RETURN
0452
0453
0454
0455   ENTRY B_SBI (LUN)
0456
0457
0458
0459   call header (lun)
0460
0461   if (
0462     1 lib$extzv(24,8,emb$1_hd_sid) .eq. 255
0463     1 or
0464     1 lib$extzv(24,8,emb$1_hd_sid) .eq. 1
0465   1 ) then
0466
0467   if (emb$w_hd_entry .eq. '07'x) then
0468
0469     call logger (lun,'ASYNCHRONOUS WRITE')
0470   ELSE
```

```

0471
0472   call logger (lun,'SBI FAULT')
0473   ENDIF
0474
0475   else if (lib$extzv(24,8,emb$hd_sid) .eq. 2) then
0476
0477   if (emb$w_hd_entry .eq. '07'x) then
0478
0479     call logger (lun,'WRITE BUS ERROR')
0480   endif
0481   endif
0482
0483   RETURN
0484
0485 END

```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	961	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	294	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	324	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 SYECOM	44	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	2135	

ENTRY POINTS

Address	Type	Name	Address	Type	Name
0-0000032F	B_SBI		0-00000000	SBI	

VARIABLES

Address	Type	Name	Address	Type	Name
4-00000012	L*1	CP_11750	4-00000011	L*1	CP_11780
4-00000013	L*1	CP_11722	4-00000014	L*4	CRYPTK FLAG
4-0000000D	I*4	DEV_CHAR	2-00000000	L*1	DIAGNOSTIC MODE
3-00000000	I*4	EMBSL_HD_SID	3-00000004	I*2	EMBSW HD ENTRY
3-0000000E	I*2	EMBSW_HD_ERRSEQ	4-0000001E	L*1	END VALUE
4-0000001D	L*1	EOF_FLAG	2-00000010	I*4	ERROR PC 750
3-000000A4	I*4	ERROR_PC_780	3-00000014	I*4	ERROR_PSC_750
3-000000A8	I*4	ERROR_PSC_780	2-00000004	I*4	FIELD
4-00000004	L*4	FORMS	2-00000008	I*4	I
4-0000000C	L*1	LINES	4-00000027	I*4	LSTLUN
AP-00000042	L*1	LUN	4-0000001F	I*4	MAILBOX_CHANNEL
4-0000002B	CHAR	OPTIONS	4-00000008	L*4	PRINTER
4-00000000	I*4	RECCNT	4-00000023	I*4	RECORD SIZE
3-00000014	I*4	SBI_COMP	3-0000001C	I*4	SBI_ERR

J 1
16-Sep-1984 00:28:09
5-Sep-1984 14:22:11VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]SBI.FOR;1

Page 7

ST
VO

3-00000010	I*4	SBI_FAULT
3-00000020	I*4	SBI_TO
4-0000001A	L*1	VALID_CPU
4-0000001C	L*1	VALID_TYPE

3-00000018	I*4	SBI_MAINT
4-00000019	L*1	VALID_CLASS
4-0000001B	L*1	VALID_ENTRY
4-00000018	L*1	VOLUME_OUTPUT

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMBSQ HD TIME	8	(2)
3-00000010	I*4	MEMORY_REGISTERS_UV1	20	(0:4)
3-00000064	I*4	SBI_REGA	64	(0:15)
3-00000024	I*4	SILO	64	(0:15)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-00000049	10'	1-0000005F	28'	1-000000C1	30'	**	50	1-000000EA	52'
**	54	1-0000010A	55'	**	80	**	85	1-00000101	53'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
	CLASSIFY	I*4	COMPRESS4		FRCTOF
	HEADER	I*4	LIBSEXTZV		LINCHK
	LOGGER		MEMORY_REGISTER_UV1		SBI_COMPARATOR
	SBI_ERROR		SBI_FAULTREG		SBI_MAINTENANCE
	SBI_SILO		SBI_TIMEOUT		VAXPSL

COMMAND QUALIFIERS

```
FORTRAN /LIS=LIS$:$BI/OBJ=OBJ$:$BI MSRC$:$BI
/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19
```

COMPILE STATISTICS

Run Time:	3.86 seconds
Elapsed Time:	11.88 seconds
Page Faults:	179
Dynamic Memory:	196 pages

0154 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SBI LIS	TIMCMP LIS	TUTAPE LIS	UBAERR LIS
STSEVENT LIS	TRNS.BITS LIS	UBA LIS	UNDEFINED LIS
	TIMRB LIS	TU81SENSE LIS	UNKN DISP LIS
	SYSPWRFL LIS		
SUMMARY LIS			
SHRVECTOR LIS	SYSTARTUP LIS	UBAINT LIS	UNKNOWN LIS
	TOF LIS		
	TSTAPE LIS		